Rule DAS620: The number of data buffers should be increased

Finding: CPExpert noticed that Non-Shared resources (NSR) was specified for VSAM data sets and most of the access was sequential processing. However, relatively few data buffers were assigned to the data sets. Consequently, I/O processing was inefficient. This finding applies only if SMF Type 42 (Data Set Statistics)[1] and SMF Type 64 (VSAM Statistics) records are available in a MXG performance data base.

Impact: This can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of applications referencing the VSAM data. The level of impact depends on the number of direct I/O operations that are used.

Discussion: A VSAM file structure consists of one or more *Control Intervals (CIs)* and one or more *Control Areas (CAs)*.

- A **Control Interval** is a continuous area of direct access storage that VSAM uses to store logical records. When a logical record is read from direct access storage, the entire Control Interval containing the record is read into a VSAM buffer in virtual storage. The desired logical record is then transferred from the VSAM buffer to a user-defined buffer or work area. While logical records within a Control Interval may vary in length, all Control Intervals in a specific VSAM data set are of the same length.

- A **Control Area** contains one or more Control Intervals. The Control Intervals are grouped together into fixed-length contiguous areas of direct access storage. A VSAM data set is composed of one or more Control Areas.

I/O buffers are used by VSAM to read and write control intervals from DASD to virtual storage. For a key-sequenced data set (KSDS) or variable-length relative records data sets (VRRDS), VSAM requires a minimum of three buffers: two buffers for data control intervals[2] and one buffer for an index control interval. Only data buffers are needed for entry-sequenced, for linear data sets (LSDS), and for fixed-length relative record data sets (RRDS).

The VSAM defaults provide these **minimum** buffers. However, to increase performance, there are parameters to override the VSAM default values.

---

[1] **%LET TYPE42DS = Y;** must be specified in USOURCE(GENGUIDE) must be specified in USOURCE(GENGUIDE) or USOURCE(DASGUIDE) to advise CPExpert that TYPE42DS is available.

[2] One of the data buffers is used only for formatting control areas and splitting control intervals and control areas.

Selecting good buffering options can reduce the number of I/O operations, reduce job elapsed time, reduce CPU time, device reduce connect time, and reduce device disconnect time. IBM benchmarks[3] have shown that, depending on the workload, selecting optimal buffering parameters can provide 90% reduction in the number of I/O operations, provide over 65% reduction in job elapsed, provide over 40% reduction in CPU time, and provide over 50% reduction in device connect time. These remarkable savings were achieved simply by altering the buffering characteristics of jobs processing VSAM data sets.

The optimum buffering techniques vary, depending the buffering techniques (or resource pools) used and how the records are accessed (sequential or direct). There are four types of resource pools, depending on the type of data sharing that is implemented:

- **Non-Shared Resource (NSR)**. NSR is the default VSAM buffering technique. With NSR, VSAM buffers are not shared among VSAM data sets, and the buffers are located in the private area. VSAM data sets with NSR buffering can be accessed sequentially or direct (or both). However, NSR is suited for sequential processing because, if the data set access is sequential, the buffers are managed with a read-ahead algorithm. The read-ahead algorithm provides overlap of I/O and CPU processing and is efficient for sequential accesses. Since NSR is oriented toward sequential access, there is no expectation that a record will be re-used (as might exist with direct processing). Consequently, once a record is processed from the NSR buffers, the buffer is likely to be reclaimed for another record read from DASD.

- **Local Shared Resource (LSR)**. With LSR, VSAM buffers normally are shared among VSAM data sets accessed by tasks in the same address space. Since LSR is oriented toward shared (and direct) access, there is an expectation that a record might be re-used. Consequently, buffer management algorithms retain buffers as long as possible, using a least-recently used (LRU) algorithm, after a record is processed from the LSR buffers. There is no read-ahead algorithm with LSR, and there is no inherent overlap of I/O and CPU processing. LSR is appropriate for direct access of VSAM data sets, regardless of whether the data sets are shared.

- **Global Shared Resource (GSR)**. GSR provides serialization of shared resources across multiple systems. In a GSR complex, programs can serialize access to data sets on shared DASD volumes at the *data set level* rather than at the *DASD volume level*. A program on one system can access one data set on a shared volume while other programs on any system can access other data sets on the same volume.

[3] *VSAM Demystified*, SG24-6105, Section 2.6.9 (Buffering options)

- **Record Level Sharing (RLS)**. VSAM RLS provides multisystem sharing of VSAM data sets in a parallel sysplex, using cross-system locking. CICS is the exploiter of RLS. RLS enables VSAM data to be shared, with full update capability, between many applications running in many CICS regions. With RLS, CICS regions that share VSAM data sets can reside in one or more MVS images within an MVS parallel sysplex.

As described above, NSR is best used for applications that use sequential or skip sequential as their primary access mode. When data sets are accessed sequentially, performance can be increased by specifying multiple buffers for the data component[4]. When there are multiple data buffers, VSAM uses a read-ahead function to read the next data control intervals into buffers as buffers become available. On the other hand, more buffers than necessary might cause excessive paging or excessive internal processing. There is an optimum point at which more buffers do not continue to improve performance.

Please note that additional buffers do not improve performance if the VSAM data set is processed with direct access. This is because there is no read-ahead I/O activity.

After applying the screening criteria specified for VSAM data sets, and extracting SMF Type 64 information for those VSAM data sets, CPExpert examines SMF Type 42 (Data Set Statistics) information for the selected VSAM data sets. CPExpert uses the TYPE42DS information to compute the percent of sequential accesses to the VSAM data set, using the following algorithm:

$$\textit{Percent sequential accesses} = \frac{\textit{S42AMSRB}}{\textit{S42AMSRB} + \textit{S42AMDRB}}$$

where: S42AMSRB = Blocks read using sequential access
S42AMDRB = Blocks read using direct access

CPExpert produces Rule DAS620 when the TYPE42DS S42DSBUF variable showed that NSR was used, the percent of sequential accesses for the data component was greater than 75%, and less than 10 buffers had been assigned to the data component for the VSAM data set.

[4] Having only one I/O buffer for the index component does not hinder performance, because VSAM gets to the next CI by using the horizontal pointers in sequence set records rather than the vertical pointers in the index set. Extra index buffers have little effect during sequential processing.

The following example illustrates the output from Rule DAS620:

```
RULE DAS620:  THE NUMBER OF DATA BUFFERS SHOULD BE INCREASED

   VOLSER: D83NE2.  Non-Shared resources (NSR) was specified for the
   below VSAM data sets and most of the access was sequential processing.
   However, relatively few data buffers were assigned to the data sets.
   You should consider increasing the number of data buffers to at least
   10 buffers, and preferably up to 30 buffers.   The I/O RATE is for the
   time the data set was open.
                                                                  ..       I/O    -ACCESS TYPE (PCT)-  BUFFERS
   SMF TIME STAMP    JOB NAME  VSAM DATA SET                      ..       RATE   SEQUENTIAL   DIRECT  ASSIGNED
   17:00,14OCT1997   NETAK31.  OPSYS.NV31.J90.AOC.DSILOGP.DATA.....  40.2     100.0        0.0      2
```

While not shown in the above example, CPExpert also shows the OPEN time for the VSAM data set. This normally is the duration of the current OPEN. If %LET VSAMSMRY=Y; was specified in USOURCE(DASGUIDE), the OPEN time represents the sum of the times the VSAM data was OPEN for all TYPE64 records in the performance data base.

Suggestion: If CPExpert produces Rule DAS620, you should consider the following alternatives:

- Increase the number of buffers for the VSAM data sets identified by Rule DAS620. IBM benchmarks[5] have shown that with read sequential access, performance improves significantly if the number of buffers for the data component of VSAM data sets is increased from the default 2 buffers to 10 buffers. Performance continues to increase as more buffers are added, but the improvement was less in IBM's benchmarks, as the number of buffers increased beyond 30 buffers. Consequently, IBM recommends that the number of buffers for the data component of sequentially accessed VSAM data sets be increased from the default to 10 buffers or up to 30 buffers if space permits. This recommendation is only a general guideline, however. The optimum number of data component buffers varies according to the amount of CPU processing[6] done between each read to the data set.

[5] *VSAM Demystified*, SG24-6105, Section 2.6.9 (Buffering options)

[6] The amount of CPU processing can be done by the application accessing the VSAM data set, or can be done by work (either system or application work) executing at an equal or higher CPU dispatching priority. The CPU-I/O overlap efficiencies created by multiple VSAM buffers for NSR sequential access occur only when the application otherwise would wait on logical records. Application delay waiting for I/O can be analyzed by the WLM Component of CPExpert, or can be seen in RMF reports for the service class to which the application belongs.

There is one significant exception to this "increase buffers" recommendation; that exception occurs when SHAREOPTIONS 4 has been specified for the VSAM data set.

The SHAREOPTIONS parameter specifies how the component or cluster can be shared among users within one system or across systems. With SHAREOPTIONS 4, the data set can be fully shared by any number of users. This setting does not allow any type of non-RLS access when the data set is already open for RLS processing. With this option, each user is responsible for maintaining both read and write integrity for the data. With SHAREOPTIONS 4, buffers are refreshed at each request. Also, the read-ahead function has no effect and defer write is not used. **Therefore, for SHAREOPTIONS 4, keeping data buffers at a minimum can actually improve performance**.

Unfortunately, SMF information does not describe the SHAREOPTIONS value, so CPExpert is unable to detect that SHAREOPTIONS 4 has been specified for a VSAM data set, and thus cannot suppress this finding for VSAM data sets with SHAREOPTIONS 4.

- Alternatively, you can specify System Managed Buffering (SMB)[7] for the VSAM data sets listed. VSAM can use system-managed buffering to determine the number of buffers and the type of buffer management to use for VSAM data sets. VSAM also determines the number of buffers to locate in Hiperspace for use in direct optimization. To indicate that VSAM is to use SMB, specify either of the following options:

  - Specify the ACCBIAS subparameter of the JCL DD statement AMP parameter and specify **Sequential Optimized (SO)** for the record access bias. This technique provides the most efficient buffers for sequential application. Approximately 500K of processor virtual storage for buffers is required for this technique, defaulted to above 16 MB.

    If 500K processor virtual storage for buffers is not available with the application, consider specifying **Sequential Weighted (SW)** for the record access bias. This technique provides efficient buffers for sequential application. Approximately 100K of processor virtual storage for buffers is required for this technique, defaulted to above 16 MB,.

  - Specify Record Access Bias in the data class and an application processing option in the ACB.

[7] Please review *DFSMS: Using Data Sets* (Section 2.5.4.2.3: Processing Guidelines and Restrictions) before implementing system managed buffering.

For system-managed buffering (SMB), the data set must use both of the following options:

- System Management Subsystem (SMS) storage
- Extended format (DSNTYPE=ext in the data class)

- Alternatively, you can exclude the reported VSAM data sets from analysis. Section 3 describes how to exclude VSAM data sets from analysis. However, you should be aware that no analysis of potential VSAM problems will be performed on data sets that are excluded from analysis.

Reference: *DFSMS: Using Data Sets* (SC26-7339 for OS/390; SC26-7410 for z/OS)
Section 2.5.5.5.1 Data Buffers for Sequential Access Using Nonshared Resources
Section 2.5.4.2: Tuning for System-Managed Buffering
Section 2.5.4.2.3: Processing Guidelines and Restrictions

IBM Redbook: *VSAM Demystified* (SG24-6105)
Section 2.6.9 (Buffering options)